

需求工程中的不确定性研究^{*}

Uncertainty Research in Requirement Engineering

王拥军 何华灿 艾丽蓉 杜永文

(西北工业大学计算机科学与工程系 西安710072)

Abstract There are at least two challenges with requirements engineering. First, requirements are usually vague and imprecise in nature; Second, requirements often conflict with each other and many conflicts are difficult to identify. This paper discusses inconsistency and imprecision on requirement respectively. Then, it points out that comprehensive method should be adopted to face with natural attribution in requirement engineering. The difficulties of this kind of methods are how to limit solving domain and induce solving model by contextual knowledge.

Keywords Requirement engineering, Inconsistency, Imprecision, Generalized fuzzy logic, Revising default theory, Evolution, Context

1 引言

用户需求一般用自然语言描述,是非形式化的,而在生成确定的、形式化的需求规格时不可避免地会牺牲用户的部分需求。采取何种方法能尽量减少信息损失和不一致,从而使需求规格更接近于用户需求是软件开发中必须解决的重要问题。

起初,在软件工程中这种转化是依靠软件开发者对用户需求的理解及自身知识完成的,使得生成的需求规格有很多的隐含缺陷,给以后的软件测试和维护阶段带来了巨大的时间和资金损耗。认识到这一问题之后,《computer》于1985年4月推出专辑介绍需求工程,希望用系统的、工程化的方法来解决需求的不确定性问题。在随后十几年的研究过程中,人们发现由于需求的变化贯穿于软件系统生命周期的始终,故而需求规格应始终处于动态变化之中,这是需求工程的核心问题。需求工程不再仅仅是软件工程中需求分析阶段的扩充,它已形成独立的研究领域^[1],有自己的生命周期(尽管对阶段的划分还有争议^[2~4]),且已研制出许多辅助工具和相关手段。

需求工程的主要目标是把握软件系统的目的和外部行为,为系统分析和设计阶段生成形式化的需求规格。从这个意义上讲,需求工程是在含有潜在不一致性的非形式化需求和形式化的需求规格之间建立联系的

研究领域,必然要面对如何处理不确定性知识的问题。比较一致的看法是:需求工程采用演化的开发模型,从不完全的需求着手,受益于信息系统和软件工程两种开发风格,形成了一个将面向工程思想和系统分析方法相融合的新领域,即采用知识工程的方法来处理需求工程领域的不确定性问题。由于用户需求往往有很强的领域特征,故这种结合给低潮中的人工智能研究提供了理想的应用背景,人工智能与需求工程的结合是非常有意义的。

我们认为需求工程的核心问题是如何处理需求的不确定性,它至少包括不精确性和不一致性两类。本文在综述了模糊逻辑方法在处理不精确需求问题时的优缺点,分析了近年来为解决需求动态不一致而引入的缺省逻辑方法之后,提出了解决需求工程中不确定性问题的几点设想。

以下各部分分别讨论需求工程中的不确定性、模糊逻辑与不精确性需求、缺省逻辑和需求的动态不一致性、解决需求问题的几点考虑和结论等。

2 不确定性是需求工程的核心问题

开发软件系统最困难的部分就是精确地决定该做什么,故而建立准确反映用户需求的详细技术需求规格是软件工程中最重要的一部分,其质量好坏直接影响到了软件系统的成败,它的失误是随后最难以修正的。

^{*} 本文得到国家教委博士点基金(98069923)资助。王拥军 博士生,主要研究模糊逻辑、非单调推理和软件工程等。何华灿 教授,博士生导师,主要研究人工智能基础理论和应用、泛逻辑等。艾丽蓉 博士,主要研究软件模式设计、人工智能基础理论等。杜永文 博士生,主要研究模糊逻辑。

从软件工程提出迄今,人们提出了许多思想和方法,企图形成精确的、一致的需求规格,这种努力大多以牺牲用户的部分需求为代价,生成的需求规格不能完全、准确地反映用户的本来意愿,而且很难适应用户需求的不断变化。导致这些困难的根本原因首先在于:客观世界的半可知性,即对外部的了解信息是不完全的,这是需求不一致的根源所在;其次,事物存在与精确相对应的模糊属性,它必然会在合适的环境下表现出来,是需求的一种基本属性;其三,人类认识客观世界的能力受到自身的经验、知识和工具的影响,也会在认识过程中表现出不确定性。需求工程应当正确地面对需求的这些属性和局限,以系统的观点容纳和管理它们。

由于模糊的、非形式化的需求往往以一定范围内的弹性形式表现出来,故在不精确性需求和形式规格方法之间建立沟通的时候,恰当估计不一致需求间的妥协是一个挑战性的问题。以下给出几个重要观点:

·不一致性管理 对于众多来源不同、认识角度各异的需求来说,出现不一致是不可避免的。软件工程师(practitioner)则把不一致看成现实的一个方面,认为在大型项目中想祛除不一致是异想天开。有时即使检测到矛盾,也因为删除这一不一致会引起其它不一致的出现而导致作罢。况且,不一致还表现出有益的一面,它在将开发过程聚焦于具体方面有很大的潜力。最近已在更广泛的意义上开发管理不一致的工具和技术,研究者采用改善而不是解决的指导思想,寻求不一致的可用性,真正把握不一致性(如超协调逻辑和真值维护系统)。目前,有很多注意集中在大范围软件开发背景(Context)下的不一致管理^[5]。

·演化需求和规格 软件开发本身就带有不断演化的特性。这主要因为软件开发与维护周期可能很长,而且开发过程完成后还要长期使用,故用户对这软件系统的需求不可避免地会发生变化,因此,不可避免地要引起整个系统设计实现的改变,这一变化也就构成另一与软件开发有关的周期,它是由需求的变化所引起的,统称为软件演化。显然这种软件演化过程形成的循环周期,是在软件开发过程之外的。

·不精确性需求的表示 应当有一种形式化的方法来捕捉需求的模糊本质。一般采用模糊集合来表示不精确性概念,以表明需求某种程度被满足。需求之间的关系又是复杂多样的,要有多样的手段来处理。

现在的软件开发是市场驱动的,这使得需求工程要面对众多来源不同、侧重各异的需求,不一致情形较以往更为复杂;软件演化带来的需求动态变化也对软件开发提出了更高的要求;尤其当系统资源不能满足顾客要求的全部时,就应当根据背景知识限定讨论问题范围,归纳出具体的问题模型和规范。

需求的不完全和不精确性是产生上述问题的根本原因,也是不可避免的两个基本因素,建立于此认识之上的形式化方法应当是从需求表示到需求关系的识别、以及推理的形式基础都遵循这两种约束,即应采用新方法处理新的需求表示,克服以往方法中表示和处理方法不相适应的问题,使解决方案从整体上把握了问题的要害——如何正确面对客观世界不完全和不精确的本质特性。

3 模糊逻辑与不精确性需求

现实的需求关系是不确定的,而确定的需求关系往往不能描述现实。

现在的模糊推理,基本上是在模糊集合的真值域上进行的,对于不精确性需求的表示,可以做到显式地捕捉这种直觉(弹性)、为不精确性需求提供形式化方法、进行矛盾需求的折衷分析和识别^[6-7]。例如,需求R,此项目的设计费用应当是低的,这里“低”就是一个不精确的概念,用语言真值可以恰如其分地表示它的弹性,它采用模糊测度来反映模糊需求之间的一种序关系和需求的满足程度,并不真关心精确取值,这很好地解释了人们大致把握问题的能力,需求的关系除了一致和对立之外,还有介于二者之间的矛盾和协作关系,模糊逻辑就分别针对以上四种情况,给出了相应的解决方法,取得了一定的成果。但这些方法并没有意识到需求之间的关系实际上也是在绝对一致和绝对对立区间内连续变化的,只采用相互包含关系之上的扎德逻辑运算模型处理,有时结果和实际生活相距甚大(见Elkan的“模糊逻辑:似是而非的成功”一文^[8]),很难令人满意。

注意到这个问题之后,有人在对三角范数的研究基础之上提出了若干种逻辑运算模型。例如,有界算子、概率算子和Yager算子等等。这在一定程度上缓和了逻辑运算模型不能适应复杂环境的矛盾,但并没有找出问题的症结所在。

已有模糊推理方法的主要问题实际在于只考虑了真值的模糊性,没有注意到建立在模糊关系之上的逻辑运算是一簇连续可控的算子。因此,需求之间关系的处理就不可能得到合乎环境和实际要求的反映,结果自然难以令人满意。Ildar在文[9]中指出:若放弃与/或算子的分配律,则可突破扎德逻辑运算模型的限制,获得连续的模糊连接词,并进而给出了忽略交换律和结合律时得到的广义与/或算子示例,这在实际环境中是有意义的;何华灿教授提出的广义模糊逻辑^[10,11]是全面研究不确定性推理的逻辑,引入特征空间来说明事物之间的广义相关关系(不仅是相互包含的扎德运算模型),其逻辑运算模型是一簇连续可控的算子,这

些研究新内容使得我们可以处理在绝对一致和绝对对立区间内连续变化的需求关系,拓宽了需求关系的范围,使相应的处理建立在对需求正确的认识之上,为从整体把握不精确性需求提供了理论基础。

从以上分析可以看出,文中的“广义”是指在考虑真值模糊性的基础上,进一步研究逻辑运算的连续可控性。例如:广义相关系数 h 从 0 连续变化到 1,指明需求关系从最大相克(跃变算子)连续变化到最大相吸(扎德算子),中间经过 $h=0.5$ (有界算子)、 $h=0.75$ (概率算子)等大家熟知的逻辑运算模型,可以处理很复杂的需求关系。

4 缺省逻辑与需求的动态不一致性

需求的变化充斥软件生命周期的始终,需求工程应当采用动态机制来适应这种变化,并努力减少变化带来的大计算量。

不一致性的根本原因是系统仅有不完全的信息。因此,形式化方法研究这类问题首先要处理似乎可信的假设,使系统相对完全化。具体处理方法是多种多样的,例如:统计、封闭世界假设、信念认知和逻辑极小化等。它们的实质是用不同的准则来形式化需求中隐含的知识,使得潜在的不一致可以被发现和解决。由于假设是基于过去的经验和一般知识的,所以这类形式化方法还必须能解决由于新需求的引入所导致的平凡化问题(即能得出任何结论),并对系统修正,以期得到合理、适势的需求规格。经典逻辑的单调和封闭性使得它对此无能为力,非单调推理(尤其是缺省推理)则正好提供了可能突破的机制。

缺省逻辑是非单调方法中非常适合表示和处理不一致需求的形式化方法。首先,它很好地利用了独特的语义解释将不一致需求分在不同的扩充中,避免了平凡结果。其次,它能有效地利用信念修正理论来管理需求规格的演化过程,因而,从 1996 年在第四届泛太平洋人工智能国际会议上第一次提出用缺省表示需求的思想之后,近几年引起了广泛的注意。

然而,缺省逻辑是一个基于假设的、带有不完全信息的严格方法,本质上是静态的推理方法,从某种意义上说,不能进行关于需求变化的推理,只是在需求变化针对缺省理论发生时,其非单调行为才表现出来。以往的纯缺省逻辑不能满足随时间变化的演化需求的要求,不能适应变化(Change)是需求工程的核心问题这一最新软件开发和维护的潮流。这迫使我们考虑和分析缺省理论的修正和演化,以确保需求的变化能在严格和易理解的风范下实现,故而,研究动态的缺省理论才是达到需求工程目标的当务之急。

修正缺省理论的研究只是最近(1998 年以后)才引

起注意,它将缺省理论看作支撑若干可选观点(即扩充)的高级规格,使变化发生在这一级而不是仅修正单个扩充。因为缺省理论代表可选扩充的集合,故这样做的优点在于和惰性估值策略相结合,仅在必须的时候才做计算,使得没有必要每一次缺省理论变化时都计算扩充,这可以节省相当大的计算工作量,适应更为复杂和大型的问题需求。修正缺省理论的提出更有助于解决变化过于频繁的问题,适用于对付大规模与大范围软件需求之间的不一致性。当前已实现的两种主要动态算子修正和约减^[11],表现出很强的过程适应能力,为修正缺省理论的前景做了乐观的诠释。

5 解决需求工程中不确定性问题的几点设想

以上从需求的不一致性和不精确性两个视角分别讨论了存在的主要问题和发展趋势。值得指出的是:研究不精确性需求的各种方法很多已经考虑到不一致问题,提出了针对不同需求关系的矛盾协调办法,但这些逻辑系统本质上是静态的,不能形成随需求变化而系统动态相适应的机制;引入非单调逻辑来解决不一致的思想采取容纳和系统管理的手段,追求从宏观上把握由不完全性引起的不一致性,然而忽略了需求本身有模糊、不精确的基本属性。可以想见,任何现实中的需求都是不一致性和不精确性的统一体,应当发展一种综合的形式化方法来适应这种客观环境的要求。下面给出关于这种形式化方法的四点设想:

·信息系统中此类问题的研究一般是在模糊理论框架下用缺省思想解决不一致问题。文[13]中以可能性理论作指导,用模糊集方法表示常识知识,取得一定的进展,但其没有考虑动态变化引起的不一致和需求关系之间的连续可控性,所以,范围必然大大缩小。文[14]提出的证据理论方法给出了证据的新表示,考虑了模糊性、不一致性和消解冲突的优先级,使最终结论反映所有相关信息的组合作用,但对矛盾的综合、折衷算法固定,不具有可控性。本文认为应发展一种基于广义模糊逻辑的缺省理论,以适应需求之间关系模糊和动态不一致的要求。

·首先,应认识到由于采用了广义模糊逻辑处理不精确需求,故需求规格的修正至少有两点不同于传统信念修正。一是需求之间可以相互妥协,用不同需求关系下的平均算子可以实现这一点;二是需求之间应当存在差别(不同时期的同一需求也应作差别处理),即主体需求部分要作为知识(事实),另一部分作为信念。当然,可以设置门槛完成两部分之间的动态转化。

·其次,由于不一致需求是不可消除的,故从某种意义上讲,只能在不一致的环境下推理,应考虑超协调方法的引入,这有助于避免平凡,得出局部有用但不一

定必然正确的结论,这和人类用常识处理非全知世界所采用的方法很相近,而且,常识推理的研究已经找出了超协调逻辑和非单调逻辑的统一新形式,如超协调非单调逻辑^[15]。

最后,在理解了以上几个观点之后,容易想到这类形式方法最大的难点在于使不精确的需求假设空间更加合理。用背景知识和领域知识约束推理的范围可以提供为人接受的、符合常识的形式规格,解决由于有限资源不能满足很多需求时优先级的确定问题,避免诸如彩票悖论之类的矛盾,非事实知识^[16]就是在部分领悟人类认识认知和智能行为的基础之上,用隐含于事实之后的环境知识限制讨论范围,大大提高了软件系统的健壮性。然而,如何获取非事实知识则不是一件容易的事,牵扯到知识发现、学习和归纳等很多领域。使用各种各样的方式嵌入推理系统的假设在受限(包括显式和隐式)软件系统和应用之间架起了桥梁。因而,研究这个问题是非常必要的,也是很有现实意义的。

限于篇幅,这里只给出了大致解决问题的几点考虑,我们将在后续文章中分别和逐步地给出我们自己在这几方面的工作进展。

结论与展望 本文把知识工程的一些思想和方法引入到需求工程领域,指出特定背景下需求合适的表示方法;用常识知识帮助限制推理范围、确定优先级和认识智能行为;提出了应以广义模糊逻辑理论为基础,深入研究需求工程中的不确定性问题,推广需求规格形式化的适用范围。

参考文献

- 唐稚松. 时序逻辑程序设计与软件工程. 科学出版社, 1999
- Krasner H. Requirements Dynamics in Large Software Projects, A Perspective on New Directions in the Software Engineering Process. In: Proc. IFIP. Elsevier, New York.
- Jarke M, Pohl K. Requirements Engineering in 2001: (Virtually) Managing a Changing Reality. Software Engineering, 1994(Nov.): 257~266
- Potts C, Takahashi K, Anton A. Inquiry-based Requirements Analysis. IEEE Software, 1994(Mar.): 21~32
- Feather M, Fickas S, Kramer J. Living with Inconsistency. In: Proc. ICSE'97 Workshop, Boston, May 1997
- Finkelstein A, et al. Inconsistent Handling in Multi-perspective Specifications. IEEE Trans. on Software Engineering, 1994, 20: 569~578
- Boehm B. In H. Identifying Quality Requirement Conflicts. IEEE Software, 1996(March): 25~35
- Elkan C. The Paradoxical Success of Fuzzy Logic. In: Proc. 11th National Conf. on Artificial Intelligence. MIT Press, 1993. 698~703
- Batyrshin I, Kaynak O. Parametric Classes of Generalized Conjunction and Disjunction Operators for Fuzzy Modeling. IEEE Transactions on Fuzzy Systems, 1999, 7(5): 586~595
- 何华灿, 刘永怀, 等. 经验性思维中的泛逻辑. 中国科学(E辑), 1996, 26(1): 72~78
- 何华灿, 刘永怀, 等. 命题泛逻辑学原理. 科学出版社(待出版)
- Williams M-A, Antoniou G. A Strategy for Revising Default Theory Extensions. In: Proc. 6th Intl. Conf. on Principles of Knowledge Representation and Reasoning. Morgan Kaufmann, 1998. 24~33
- Yager R R. Fuzzy Sets Methods for Representing Commonsense Knowledge. Journal Intelligent and Fuzzy Systems, 1999, 7: 27~45
- Rocha M. Evidence Sets: modeling subjective categories. International Journal of General System, 1997, 27: 457~494
- 林作铨. 超协调限制逻辑. 计算机学报, 1995, 18(9): 665~670
- Hoffmann A. Non-Factual Knowledge. Paradigms of Artificial Intelligence, Springer, 1998. 243~252
- Rumbaugh J, et al. Object Oriented Modelling and Design. Prentice Hall, New Jersey, 1991
- Booch G. Object Oriented Design with Applications. Benjamin/Cummings, Menlo Park CA, 1994
- Selic B, et al. Real-Time Object-Oriented Modelling. John Wiley & Sons, Inc. New York, 1995
- The Object Management Group. UML Notation Guide, Version 1.0(1997-1-13)
- Musa JD. Operational profiles in software reliability engineering. IEEE Software, 1993(March)
- Runeson P, Wohlin C. Usage modelling: the basis for statistical quality control. In: Proc. of the 10th Annual Software Reliability Symposium, Denver, Colorado
- Wholin C, Runeson P. Certifications of Software Components. IEEE Transactions on Software Engineering, 1994, 20(10)
- Runeson P, Wohlin C. A dynamic usage modelling approach to software reliability engineering. [Thesis]. Department of Communication Systems, Lund University, Lund Sweden
- Wesslen A, Wohlin C. Modelling and generation of software usage. In: Proc. Fifth Intl Conf. on Software Quality

(上接第34页)

- Rumbaugh J, et al. Object Oriented Modelling and Design. Prentice Hall, New Jersey, 1991
- Booch G. Object Oriented Design with Applications. Benjamin/Cummings, Menlo Park CA, 1994
- Selic B, et al. Real-Time Object-Oriented Modelling. John Wiley & Sons, Inc. New York, 1995
- The Object Management Group. UML Notation Guide, Version 1.0(1997-1-13)
- Musa JD. Operational profiles in software reliability engineering. IEEE Software, 1993(March)
- Runeson P, Wohlin C. Usage modelling: the basis for sta-